European Commission

JRC CONFERENCE AND WORKSHOP REPORTS

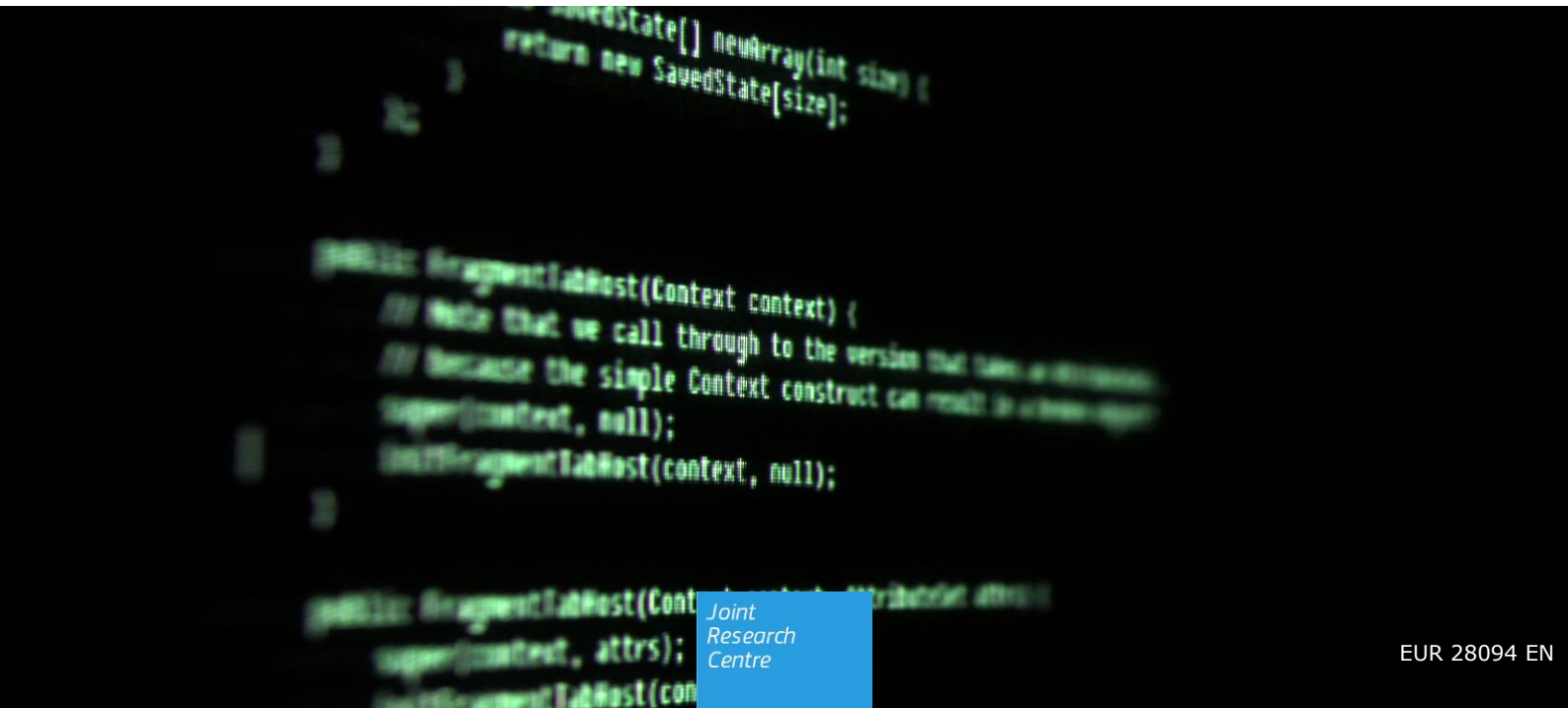
Software as an Asset for Technology Transfer

2nd TTO Circle workshop on best practices

A workshop report

Annarita Ferreri

2016



Joint Research Centre

EUR 28094 EN

This publication is a Conference and Workshop report by the Joint Research Centre (JRC), the European Commission's in-house science and knowledge service. It aims to provide evidence-based scientific support to the European policy-making process. The scientific output expressed does not imply a policy position of the European Commission. Neither the European Commission nor any person acting on behalf of the Commission is responsible for the use which might be made of this publication.

Contact information
Name:
Address:
E-mail:
Tel.:

JRC Science Hub
https://ec.europa.eu/jrc

JRC 101760
EUR 28094 EN

PDF	ISBN 978-92-79-61701-0	ISSN 1831-9424	doi:10.2760/716557
Print	ISBN 978-92-79-61702-7	ISSN 1018-5593	doi:10.2760/582578

Luxembourg: Publications Office of the European Union, 2016

Table of contents

Executive Summary

This report includes a summary of the discussions held and the presentations made at the 2015 Technology Transfer Office Circle (TTO Circle) workshop on 'Software as an asset for technology transfer', jointly organised by three of its members: the European Commission's Joint Research Centre (JRC), the European Organization for Nuclear Research (CERN) and the French Institute for Research in Computer Science and Automation (Inria).

The workshop took place in Brussels on 30 September and 1 October 2015 and is the second event organised in the framework of the TTO Circle workshops for the promotion of best practices in technology transfer. These annual workshops are held to present and discuss different approaches to technology transfer and to compare different strategies for commercialisation of intellectual property (IP).

They are also considered as the fora where participants can extend their discussions in smaller groups, and explore areas for future collaboration.

Participants benefited from the experience of the various invited speakers, a mix of TTO Circle members and outside experts from academia, publishing and patent attorney practices, as well as from the Q&A sessions where animated discussions took place.

The purpose of this workshop was to analyse the development of software products in public research institutions and its early-stage commercialisation, in order to foster thought-provoking discussions among members and define and draw conclusions on best practice, by comparing lessons learned, challenges and success stories.

Suggestions and comments made during the workshop have been taken into account to produce this document, and are based mainly on the content of the presentations in the different workshop sessions, and on the outcome of the discussions among participants. All the individual presentations can be found and downloaded from the website: https://ec.europa.eu/jrc/en/event/conference/9th-plenary-meeting-european-tto-circle?search

The workshop

The workshop agenda and a list of participants can be found in Annex I of this report. The programme comprised four separate sessions, and the report follows the same workshop structure. A summary of the plenary sessions and the discussions around these topics is given below.

The first session **'General presentation - setting the scene'** focused on examples of public research spin-offs where different business strategies had been applied. The three speakers presented case studies of successful technology transfer of software developed in their research organisation.

The second session **'Forms of software protection'** highlighted different ways of protecting software. First, the presentations explored the classic copyright mechanism, before moving on to the use of patents as a form of protection, and concluded by assessing the implications of opting to license software.

The third session on **'Open source'** explored the subject in detail as well as approaches to risk valuation and auditing. The four presentations touched on open source (OS) mechanisms and concepts, OS strategies and practices, auditing issues and tools, risks and licence compatibility.

In the fourth and last session **'Business models and spin-offs'** there was an animated discussion on business models and methodology for creating spin-offs and dissemination strategies.

The main issues discussed were also mainly derived from the case studies analysed.

Day 1

General presentation - setting the scene

Bringing a technological invention to the market usually follows two possible paths: either the said technology is licensed to a third party via a licence agreement, or a spin-off company is created.

The research institution where the invention was generated can claim the intellectual property (IP) rights on technologies developed in their laboratories. This IP typically draws on patents or, in other cases such as for software, on copyright protection. In Europe, software is protected quite easily via copyright if it is considered as a stand-alone creation, although it can also be protected with a patent, typically in connection with the use of hardware equipment.

The case of software commercialisation is of particular interest from the business model point of view. At a very high level, companies can actually opt either for an open-source licence or for a proprietary licence. In the first instance, the **source** code is made available with a licence in which the copyright holder provides the rights to:

- Use
- Copy
- Modify the source code
- Sub-licence/sell the initial software and/or the derivative software due to modifications of the source code

In the case of the proprietary model, the licensee is only allowed to use the software, but has no information on the source code and is not in the position to redistribute the product. Advantages and disadvantages of open source vis-à-vis proprietary business models were discussed in more detail in session 3 of the workshop.

In this first general session, speakers shared their experience on creating spin-off companies within their organisation and the different business strategies adopted.

Public research software dissemination – a view from the open source industry

Jean-Paul Lorre' of Linagora, a French company founded in 2000, explained that when trying to mobilise high-value investments, results should be made available to the market quickly. To accomplish this, Linagora has opted for the "free-free" model, whereby software products are offered licence free, and with no premium. Their licence model is designed to emphasise collaboration and to encourage broad use. Their main revenue stream comes from consultancy and support to clients. Mr Lorre stressed how such an approach is a facilitator for public research dissemination because it allows researchers to go on using and contributing to software results after the project has ended. It guarantees sustainability, thanks to the exchange within the 'community' of developers, and protects and facilitates broad usage of public investment.

Software start-up company – the experience of a public research spin-off firm

Next, Folkert Teernstra of TNO addressed the general importance of selecting the right team for setting up a spin-off – the right mix of competences is crucial for gaining the confidence of the investors. TNO may also invest in a spin-off after its formal establishment. Where TNO decides to hold stock in a spin-off, it distinguishes between strategic and non-strategic participation. Shares in the latter category are usually divested within a fairly short period of time. In respect of strategic participations, TNO's general policy is to maintain ownership of a majority of the shares. Sometimes a minor amount is held by key personnel. A privatised holding company may provide office spaces and basic legal and business support.

Mr Teernstra gave an example of a software package developed by TNO's microbiology department. As the software was needed for a variety of other applications, TNO chose not to disclose source code and to create a spin-off. To overcome the need for the software to be operated by specialised microbiologists, TNO opted to have patents licensed for specific 'field of use', and for modelling services to be provided under a service-level agreement (SLA), while the results would be owned by the spin-off. The software is made available to the spin-off by means of third-party-hosted SaaS service (software as a service).

TNO's revenue stream is based on the licence fees paid by the spin-off company, and which conform to market prices. The valuation is done either by benchmarking, or a cost-based approach or a market-based approach (cost of other means of exploration). The microbiological analysis is performed against (cost of other, competing means of exploration). The microbiological analysis is performed against normal fees, under the SLA. Finally, there would be profit sharing with TNO companies on the final sale of shares, in case of acquisition by a third party.

Mr Teernstra stressed that TNO, as a partly public-funded body for applied scientific research, believes that the most correct way to make use of taxpayers' money is to use these public means carefully and conservatively by charging a portion of the development cost to the downstream commercial users of the results. This is primarily done by IP-protecting these results. Subsequently, the results can be licensed against a reasonable fee that is proportional to the commercial use by each respective licensee. While from time to time TNO publishes code in open source, he also emphasised that this requires a careful approach. Open sourcing results obtained with public money, where closed source commercial alternatives exist, could easily distort the market.

Following this address, Asier Rufino of Tecnalia Ventures tackled the subject of how to maximise on commercialisation opportunities.

First of all, he proposed a different view of the 'valley of death'. For the Tecnalia team there is a gap in the pool of talent available to lead and manage a spin-off/start-up. Not all scientists and inventors feel inclined to take on a more managerial role, often preferring to continue with their research activities or stepping into the role of CTO. Finding the correct team is pivotal for the future of the start-up; any platform able to support scouting for entrepreneurial talent would be a key success factor in mobilising even cautious investors. At Tecnalia, they also believe in having an open communication channel with potential future customers in order to help identify the challenges that most limit their profitability potential.

The Tecnalia approach is three-fold: search for minds who can use technology to offer a customer-value proposition based on the ability to solve large profit and loss problems to a clearly defined market segment with a global reach. Then, set up of the correct management team, with an entrepreneurial spirit and willing to convert disruptive technologies into ready-to-invest business opportunities. And last but not least, identify investors seeking to transform technology-based investment opportunities in business

value and growth. In fact, they prefer to bring the investors in as soon as possible to shape the product's development process.

In practice, they advocate technology and know-how that are transferred to the spin-off along with an executive team employed directly by the spin-off with equity-stock-options-ratchet schemes but no right to return to Tecnalia. Finally, they always define a clear exit strategy for Tecnalia (and hence for the investor).

They foster the use of a knowledge portfolio. Their main policy is to share best practices, knowledge and expertise. To do so, they must establish informal channels of communication with policy-makers and organise training programmes. In this way, they can develop a common approach towards international standards for the professionalisation of technology transfer procedures.

Mr Rufino presented two case studies of different business strategies adopted to launch two software companies to be spun off from Tecnalia. The first one was a wearable technology enabling guest profiling/tracking. To launch the product, they chose to set up a partnership with Onity, the leading global provider of electronic locks for the hospitality sector, with more than 30 000 hotels.

The second one was a cloud-based game platform that allows developers and brands to create, publish and monetise HTML5 games with ease, since no programming is required. They opted to organise a more classic funding event – including business angels – and in its final round of investment, the company secured a substantial financial sum (~ EUR 400 000).

Session 1: Forms of software protection

During the last few years we have experienced spiralling growth in software and mobile applications, which has reshaped the technology landscape. In particular, software developers and technology companies have to face the complex decisions of selecting the best mechanisms to first protect and then distribute their software assets in order to maintain their competitive edge.

This session focused on understanding the legal tools available to protect software, and the impact of the choices made on the specific business case, with particular focus on the implications for licence agreements.

Copyright as means of protecting software

After lunch, Jean-Paul Triaille, IP legal officer of the JRC TT office, introduced the subject of the available forms of software protection. Software is time consuming, expensive and complex to develop, but very easy to copy, thus legal protection against copying is required. Software is hybrid in nature, since it is 'written' like a text, but it can also run/command machines and industrial chains, such as a piece of machinery. Consequently, there are two possible ways of protecting it: copyright for the 'text' and a patent for the 'piece of machinery'.

Copyright protection only extends to expressions and not to ideas, procedures and operational methods or mathematical concepts as such, as is the case for a patent, to a certain extent. Thus, copyright law will not readily prevent the creation of a competing program based on the same concepts developed in a comparable, existing program.

In the US, it is often considered easier to opt for patent protection for a computer program than in Europe, although the situation is evolving.

Clearly, patent protection can be hard and expensive to obtain but it offers greater protection, as it may prevent others from making, using, selling or importing the patented invention. However, if it is not connected to a piece of equipment, in principle it is not easy to issue, and the developer will have to opt for copyright protection.

Obtaining copyright protection is easy: in countries party to the Berne Convention for the Protection of Literary and Artistic Works (the Berne Convention) copyright protection does not depend on compliance with any formalities such as registration or the deposit of copies. It can also last between 50 and 70 years after the death of the developer (in the EU). In addition, copyright does provide some protection against non-literal infringement, such as the creation of 'cloned' software, but is better suited to preventing the complete duplication of a software program, as well as copying a portion of the software code (both of which are examples of 'literal copying').

Patents as a means of protecting software (including comparing EU vs. US)

From the discussions, it became evident that software patents might provide much broader protection to software developers than copyright law. In the second presentation in this session, Frank Van Coppenolle, patent attorney and head of GEVERS High-Tech Patent Team, raised the question of whether the prejudice "software is not patentable in Europe" is justified, or rather is obsolete and made redundant by the evolved case law. He also argued that there is clearly a convergence with American patent practice.

Mr Van Coppenolle affirmed that there are huge opportunities for technology transfer offices to create value via patents in Europe and the United States. He attempted to demonstrate the criteria of a patentable invention according to the EPC (European Patent Convention), explaining that he had learned to look at an innovation from a new angle which could result in a novel approach to the decision as to whether to opt for a patent or a copyright on a software product.

First, he quoted the EPC Art 52(1): "European patents shall be granted for any inventions, in all fields of technology, provided that they are new, involve an inventive step and are susceptible of industrial application". Based on this article in European patent law, he emphasised the requirement that the subject matter must be of a technical character, which is not always the case, prima facie, for software inventions.

He also highlighted that Art 52(2) of the EPC does not de facto include a definition of what an invention is, but rather excludes subject matter which cannot be considered as an invention, including computer programs.

However, according to Art 52(3) of the EPC: "Paragraph 2 shall exclude the patentability of the subject matter ... to the extent to which a European patent application … relates to such subject matter or activities as such". Therefore, he argued that under Art 52(2)(3) of the EPC, only a few computer-implemented inventions are excluded from patentability. This is confirmed by a recent Enlarged Board decision which follows the earlier applied COMVIK approach (T 0844/09): " … an invention consisting of a mixture of technical and non-technical features, and having technical character as a whole, is to be assessed with respect to the requirement of inventive step by taking account of all those features, which contribute to the said technical character, whereas features making no such contribution cannot support the presence of inventive step".

He concluded that more often than not it is arduous to separate an invention into strictly technical or non-technical features. An invention may have technical aspects which are embedded or even hidden in a largely non-technical context. He therefore concluded that more patent applications could be filed for software than most people think. The novelty of an innovation is in the technical solution it offers to a technical problem. If this technical problem is not solved in the same way to achieve the same effect in the pre-existing state of the art, under current EPO practice, a computer-implemented invention should be patentable at the EPO.

Copyright or patent: what implications for licence agreements?

The views expressed by Mr Van Coppenolle naturally led to the next potentially critical question: if more patent applications are filed for software rather than counting on copyright protection, what would be the implication for licence agreements?

David Mazur, IP Dissemination Section Leader in CERN's Knowledge Transfer Group, set out to provide some insight and personal views on the matter. He provided a number of circumstances where patent-related questions could arise under a software licence agreement: in the exceptional situation of a patent on a computer-implemented invention, in the more frequent case where the software will be used for a product or process covered by a patent, or where third-party patent rights had to be considered. However, he cautioned that both the economic viability and the legal obligations towards a third party should be carefully considered.

He then gave an overview of various issues to be considered when licensing patents under an agreement, taking into account that the decision to patent comes at a cost, due to prosecution and maintenance costs, which will vary according to how wide the geographical scope of the patent applications will be.

Such a decision is possible and should be carefully gauged to strike the correct balance between patent costs and licence fees (positive and negative). In particular, he invited the audience to reflect on how a profitable business model can be constructed if the software is open source (OS) and has no licence fee. Patents need a cost-recovery mechanism, but OS might affect the rentability power of the said product. If opting for a patent, it might be necessary to renounce the idea of OS in favour of a more stringent agreement that would safeguard the commercial potential in its entirety.

In which case, does that mean that OS and licensing are mutually exclusive? Other important points to consider would certainly be patent clauses in OS licences and their commercial repercussions. It is not inconceivable to foresee a scenario whereby a conflict might arise between OS licences granted (maybe even without a person's knowledge) and patent licensing, or, for example, cases when the recipients of the OS software do not like the copyleft clauses imposed on them.

Mr Mazur also reflected on the normal complications of joint ownership, and on the fact that a strategy should be in place to manage the added complexity that may arise from licensing patents under a software licence agreement compared to what usually happens with copyrights. Would the standard licence safeguards still be applicable – such as freedom to abandon patent rights and the duty to inform the counterpart of any decision?

Patent retaliation under GPL v.3 Article 10 – Automatic Licensing of Downstream Recipients states that: "… you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the program or any portion of it".

This can be a very difficult clause to apply if the costs have to be recovered.
From CERN's point of view, public research should have a social impact and should maximise the technological and knowledge return to society, especially to consumers in the funding Member States. It also has the function of promoting the image of the organisation as a centre of excellence for technology. Both these aspects must be accounted for when preparing a strategic roadmap for how best to exploit the inventions.
Mr Mazur concluded his address by stating that patenting is an opportunity that comes at a cost and it needs to be carefully evaluated in the framework of social impact, too.

Day 2

On the second day, the presentation focused on open source (OS) and its impact on business strategy, and on some general business models for software commercialisation.

Session 2: Open source

The four presentations that followed in Session 2, all seemed to reach the common conclusion that it is important for TT officers, lawyers and developers to engage as early as is feasible in the process of software development, in order to facilitate the design of the commercialisation roadmap.

Open source mechanisms and concepts

Jean-Paul Triaille, from the JRC TTO, gave an overview of OS software licensing.

The issue of access to source codes in software licences is often handled explicitly and source codes are either made available or are put in escrow of a third party, or they are not made available.

If the agreement does not state anything, there is no right of access for standard software. On the other hand, for custom software, judges usually accept that there is right of access if this is necessary for the continued use of the software. Mr Triaille stressed that it is important to distinguish between access to source codes and the transfer of intellectual property rights: the former does not imply the latter.

Computer programs can be licensed under different logic/business models, most commonly:

- The "proprietary/commercial/closed" logic, in which the user will only benefit from rather limited rights of use;
- The "free software/open source" logic, in which, on the contrary, the user will benefit from very substantial rights of use (the so-called "user freedoms").

A copyright licence agreement is like all other types of agreements: it includes both rights and obligations. However, it is important to note that at present there is no legal regulation on free software licences (OSS) (except for some public tender regulations).
The four basic rights provided for in OSS licences all provide for the "four freedoms" for users:

- Freedom to use, in whatever context;
- Freedom to copy;
- Freedom to modify;
- Freedom to distribute.

With these rights, there are also obligations: disclosing the source code; keeping the copyright notice and the disclaimers in the code; granting the same rights to users downstream, if the OSS licence is 'copyleft'; and no right to claim royalties for use (although it is possible to ask payment for services).

There are many sorts of OS software licences, with different levels of copyleft effect. Naturally, the business models will have to be crafted on the basis of the type of licence selected.

For example, different problems in the further exploitation of software might be encountered. If the licensee only uses the software for his/her own internal needs, he/she does not have to disclose the source code or abide by the copyleft clauses. However, this will be necessary if he/she integrates copyleft OS into his/her own software and wants to distribute the latter to customers or clients.

This will have a number of repercussions: for example, in the case of a future sale/spin-off, if the envisaged business model is based on proprietary licensing, then the investors should audit the code and search for OSS copyleft. Should they find any, they will have to further examine the impact of this on the new company's future commercial activities and may be well advised, for instance, to negotiate adequate warranty clauses.

Therefore, Mr Triaille advocated that decisions with a commercial impact should not be left to developers alone; the roadmap should be designed upstream and should be the result of a triangular process of consultation among the commercial experts, the legal advisers in a TT Office and the software developers. Logically, an OS policy should be drafted and well understood and applied by everyone, starting with the developers.

His presentation provoked a number of interesting questions, not only on the best licensing strategies, but also on the commercial viability of the OS model and the role of auditing as an upstream best practice. These subjects were further explored by the speakers who followed.

Open source strategies and practices

In his presentation, Benjamin Jean, CEO of Inno3, a company founded in 2001, addressed how the OS model has changed the way the world looks at innovation protection and management. Before engaging with the OS model, it is necessary to know and understand its principles as it involves not only licenses but also a genuine cultural shift. In order to fully benefit from this change, Mr Jean argued that better IP rights and communities management are required. In fact, communities are the essence of the OS approach, enabling the real value of a product to be achieved and bringing a direct counterpart to the initial project.

Mr Jean asserted that choosing an OS strategy does not reduce the software's commercial appeal. On the contrary, not only does it have a positive impact on sharing but it also encourages knowledge creation, and by doing so, fosters innovation. Likewise, OS is a way to enforce interoperability and to create de facto a standard[1].

Technology transfer offices are usually cautious, if not reluctant to follow the OS route, since OS is often perceived as limiting a product's profitability. However, Mr Jean claims that OS can accommodate the technology transfer process for software commercialisation, so the question is: why and how?

[1] Regarding open source and interoperability information licensing, see The INTILA licence template (Interoperability Information Licence Agreement): final report, Philippe Laurent, Benjamin Jean: http://bookshop.europa.eu/en/model-contracts-for-licensing-interoperability-information-pbKK0414809/

On this subject, Mr Jean recommended certain best practices that support the OS initiative while safeguarding both the market and the software's innovative potential. First, he advised avoiding OS washing – a software or company that claims to be open source but is not – which reduces users' full freedoms. For instance, as regards licences, it is important to be aware of 'abuse of exceptions' practice as it may increase users' obligations and therefore render its application even more complex. As concerns governance, it is strongly advised to draft a contributor agreement where only specific individuals or entities can commit/contribute.

In any case, it is crucial to clarify – in advance – the IP management approach within a project. The OS licence may include clauses covering trademarks and patents; these clauses must not limit rights to use, modify or distribute the software. Furthermore, separate patent and trademark policies can complete the OS licence.

Another suggestion was to define an OS strategy and to introduce a structured governance approach. Governance ensures the effective implementation of the OS policy (which formalised the OS strategy) by assigning specific responsibilities to different bodies, so as to guarantee that clients, suppliers and/or OS communities would collaborate efficiently (for instance, commit to making a contribution).

Every organisation integrating free software/OS must ensure that the various associated licences are respected. Within the framework of the industrialisation of such use, it is necessary to implement a dedicated policy, in line with existing policies for intellectual property and quality. Mr Jean considered that the redaction of several structuring documents is necessary for immediate use by different departments and is apprehended under multiple issues (particularly industrial, technological, legal and human resources).

Members of the audience agreed on the need to redact a set of policies for OS – something that is still lacking – which must be harmonised at the global level. Otherwise, there is the definite risk that the OS approach might remain a collection of uncoordinated strategies, separately formulated and implemented by the single commercial actor.

Auditing issues and tools

Stephane Dalmas, head of industrial partnerships at Inria, explored the subject of auditing in greater detail.

It became apparent from the previous two presentations that any software can benefit from auditing, to ensure the robustness of any license agreement connected to it, whether proprietary or OS.

Mr Dalmas emphasised that for public research organisation (PROs), OS software is a complex issue: there are many existing components used in software developed by a PRO, some proprietary to the PRO, and some which have to be licensed in. Research often has to be funded externally, which means that part of the software might be developed under contracts/funding schemes covered by certain specific legal obligations (research contracts with an industrial partner, consortium agreements for collaborative projects like EU-funded ones). Some products might become joint ownerships with industry or other academic and/or research institutions, and there might be cases of code reuse (modifying, rewriting, translating, etc.).

Freedom to operate and economic value may be very difficult to determine in this context, which might limit, or even hinder the use of the software with external (industrial) partners. It might also have negative impacts on the value of any technology transfer (for example, the value of a spin-off based on a given software).

Hence, his message was clear: there is a need for better due diligence and audits in order to check compliance with a technology transfer project and exploitation plan. In particular, an audit methodology is required to understand the software's legal situation.

Mr Dalmas reinforced the point made in previous presentations, and explained the task force required to perform the audit should comprise a TT manager, an IP lawyer and the developer/researcher. The task force should operate upstream and would profit from the various competencies/know-how/expertise required to clearly assess the case:

- Understanding the code and its history;
- Understanding what is in the contracts;
- Deciding what should be achieved in term of technology transfer.

Inria has developed an IPR-tracking methodology – QualiPSo – within the framework of an EU-funded project on the quality of OS. Naturally, the tool per se is not a guarantee of freedom. Rather, it is used to improve the due diligence process (best practice).

This method helps to define the legal status, to check compliance with the TT and the exploitation plan, and to design a global strategy.

Mr Dalmas then went on to explain the steps involved in Inria's methodology which have been documented in the presentations available on the TTO circle website.

The copyleft paradox: open source compatibility issues and legal risks

Stefano Gentile, one of the software copyright specialists at the JRC's TT office expanded on the risk of incompatibility between licences due to the copyleft mechanism. In principle, whilst the adoption of copyleft components is not per se a limiting factor in the development of software, it is also important to understand that there might still be limitations during the distribution phase.

In particular, the issue of copyleft cross-licensing is both legal and technical at the same time. Mr Gentile's presentation started from the so-called 'viral effect', which stems from a subset of OS licences known as copyleft licences. He proceeded to illustrate how the unplanned combination of different copyleft licence types may generate conflicting obligations during the downstream distribution of the incorporating software.

He highlighted the paradox inherent in the copyleft obligation: on the one hand, it is designed to promote sharing and cooperation by forbidding restrictions to the redistribution of code. On the other hand, if combined with another copyleft licence, it may trigger the opposite effect by creating legal barriers to the redistribution of the resulting code. Under these circumstances, OS licensing clearly ceases to be an asset for technology transfer.

Like the previous speakers, Mr Gentile encouraged developers to cooperate as early as possible with TT offices in order to craft the best possible product. The potentially adverse effects of OS copyleft licences should be assessed upstream, during the design of the software product. In fact, often during this phase, OS components are identified and selected as incorporating elements of the software.

In addition, he recommended formulating the preferred distribution strategy as early as possible so as to establish which OS licences should be adopted to avoid any interference with the chosen distribution mechanism.

Finally, Mr Gentile reminded the audience that careful consideration of the copyleft linked to a software is not a matter of drafting the best commercial strategy – it actually underestimates its impact from a legal point of view. Under established case law, disregard for the applicable OS licence conditions (e.g. acknowledgement) is considered an outright copyright infringement.

Session 3: Business models and spin-offs

The next and final session focused on the analysis of business models for software commercialisation. Different speakers offered insights, highlighting what, in their view are the most relevant considerations to make, when embarking on the challenging and exciting process of bringing research outputs to the market.

Proprietary vs. open source business model: how to choose

The speaker at the opening session was Jérémie Fays, Technology Transfer officer at the University of Liège. He compared the option of following an OS-source strategy with a proprietary route, which can vary according to the specific circumstances underpinning the particular business case under consideration. He illustrated his talk with some examples, sharing lessons learnt for future reference when faced with the decision to choose one direction over the other.

First of all, he reiterated how the two business models are embedded in two different, almost philosophical approaches to divulging a scientific discovery: on the one hand, developers keen to follow the OS doctrine and to increase free exchange and mobility between researchers; on the other hand, TT offices industriously trying to mitigate the financial, legal and business risks associated with going OS (no license fees for cost recovery; competitors, who could provide services; competitors who could copy ideas, etc.).

In Mr Fays' view, the natural starting point should be less dogmatic and more practical: it is important, first and foremost, to frame the commercial opportunity correctly and evaluate what is at stake for both the client and the inventor. The two business models are not necessarily mutually exclusive; in fact, he emphasised, if anything, they could be rather complementary.

The classic case is the dual-licensing model, which consents compatibility by introducing a free, open licence for the basic software version, and a fee-paying licence for the enhanced pro version/s.

Other possible revenue-generating alternatives to consider for OS software are 'sponsoring', as in the case of Firefox, downloading fee-paying subscriptions, and donations such as in the example of Wikipedia. At any rate, any OS can be accompanied by the offer of fee-paying services, such as an audit and consulting. Tailored customer development and the training and certification of authorised partners should also be considered.

Furthermore, added value can be captured by releasing extra products, like documentation, additional software, and hardware to connect to smartphones, etc.

One interesting option discussed was the possibility of so-called 'customer pooling', where it is possible to 'pool' customers to share costs of any proposed development.

In Mr Fays' opinion, there is only one instance when OS should be mandatory – in the case of the public sector – since he believes that such an approach provides both the developer and the customer with added value. The customer can test the software before buying it, thereby avoiding any lock-in, while the developer benefits from receiving a quality audit, thanks to the participation of the 'community'. He also conceded that all these questions can sometimes be very theoretical because when the research team wants to follow an OS it can be hard to convince them otherwise. This view was opposed by members of the audience who were more in favour of introducing a

formal governance process, whereby the decision is triangular, among business experts, technologists and legal advisors, in order to maximise the product's value.

Business models

The last two presentations from Inria focused on the steps to take to structure a business model for software.

Arnaud Laprevote is currently working in the transfer and valorisation service division of Inria, although in the past he was the developer behind the creation of Linbox FAS, a French start-up. He began his story, including all the key positive and challenging moments of his direct experience as an entrepreneur, from his time as a software developer to becoming the CEO of his own company.

Linbox FAS was one of the very first open source service companies in France in 1996, and had its first round of investments in 2001. It then developed a pc management solution for organisations. It was bought by Mandriva in 2007; Mr Laprevote became the company's first research director then, three years later, Mandriva's CEO for 15 months before moving to a new role in a new company.

He compared the software world to the movie industry, since both sell intangible goods. However, in the film industry, releasing a movie is the project's endgame, whilst releasing software is the beginning of a long journey, with no clear end in sight. All software becomes obsolete as soon as it enters the market, and it needs continuous development.

Some consider the unfinished aspect of software is the ultimate argument for pursuing the OS route. However, in Mr Laprevote's view, it is important not to discount the fact that software editors need steady revenues as their costs are recurring, and revenue generation is closely linked to intellectual property rights.

This consideration should taken into account for in-depth reflection on strategies to incorporate the OS approach with proprietary business models, as they can be compatible.

He recalled that Mandriva, in addition to selling Linux distributions through its online store and authorised resellers, also decided to create and sell subscriptions to the Mandriva Club. There were several levels of membership with different benefits attached and costs ranging from USD 66 or EUR 60 per year (as of 2007) to EUR 600 per year.

Club members were given access to the Club website, additional mirrors and torrents for downloading, free downloads of its boxed products (depending on the membership level), interim releases of the Mandriva Linux distribution, and additional software updates. For example, only gold-level and higher members could download Powerpack+ editions.

Although many Mandriva commercial products came with short-term membership of the club, Mandriva Linux was completely usable without club membership.

His recommendation was to be imaginative and consider the options at all times: many possible revenue streams are always available, and each should be considered and weighted. It is important to stretch the company strategy by formulating business hypothesis beyond the more traditional models. And attempts should also be made to re-frame the opportunities connected to a familiar platform, such as for example, the tested dual-licensing agreements, in order to push boundaries.

Support for spin-offs

Stephane Dalmas, head of operations in the Technology Transfer and Innovation Department at Inria continued to provide illustrative examples and to share lessons learned about software commercialisation. He focused in particular on the case of software developed in research institutions and the creation of a spin-off.

Creating a spin-off from a research institution is increasingly becoming real business for a TTO. A lot of expertise must be invested to maximise its value. Spending money is needed to 'mature' promising technologies and sometimes even to educate the people inside the organisation on finding the right external people (CEO, business people), working with investors and other professionals (business schools, incubators, alumni networks, etc.).

Software-based spin-offs have their own specificities. Relatively low investment can result in a viable company, although the initial conditions can change very quickly (business models, people, ownership, location, etc.). This is why software is a complex object both from the technical and the legal point of view.

As complicated as it might sound, Mr Dalmas said he believed it is still possible to develop software in a research institution and successfully commercialise it through viable monetisation routes, sometimes even when it is released for free.

The focus of his address was not on the business model per se, but rather on the internal organisation of a spin-off.

When any spin-off is created, its aim is to promote commercially a given technology product. In the case of a software package, extra complications must be taken into account.

In some cases, it might be one of the, if not the main research tool of the unit which developed it in the first instance. If so, it is necessary to consider how commercialisation might affect the continuity of the existing line of research and the staff involved.

In addition, its intangible and continuously evolving nature means that it cannot be considered as a static, stand-alone invention, but rather as a fluid product, often needing to be further developed and integrated in an external environment (whether it is hardware or software). Having the correct team in place is always important for any business. In the case of a software company, Mr Dalmas believes that the occurrence of recurrent modifications means that strong business acumen alone is not sufficient; senior management must be hands-on and have an intimate knowledge of the product. In other words, the correct balance must be struck between the desire to retain good staff inside the institution while guaranteeing that knowledge is transferred from the staff to the spin-off.

So, how will the resulting spin-off impact the 'business' of the team from which it originates? First of all, it is necessary to understand and decide the future of the transferred software product, with respect to the researchers who first developed it and the community that will use it. Consideration should also be given as to whether it is better for the spin-off to stay inside the institution or for it to detach itself from it. Clarity and good vision on the possible evolutions of the original business model are required in order to make the correct assessment. Once the first line of the revenue-generating scheme/s has been established, flexibility should be maintained in order to define/refine what is on offer with different clients and users according to their needs.

Sometimes, leaving the institution is the best and simplest solution. If a team can do without its support, separation will bring greater freedom. Such a solution might also be crafted so as not to disregard the need to take care of the community. For example, cloning is a potential solution whereby two versions of the software package will exist, with certain differing exclusivity features. In this way, everybody's business is preserved, although the trade-off represents a loss of value for the spin-off.

Another solution is shared custody. However, in this case, a common temptation is to give the spin-off the role of maintenance, with the researchers from the institution producing new versions/features. Here, the commercial drive could be compromised and the entire operation put in jeopardy.

Even with OS, it is possible to licence either a less 'restrictive' licence (when GNU GPL), or some 'proprietary' components (not previously open sourced), or offer paid consultancy services in exchange for the know-how of the researchers. Last but not least, it is always possible to file for trademarks. Some of these solutions work even for joint ownership.

One important decision to gauge is whether to accept external contributions; this does not work by default but really depends on the characteristics of the product on offer. For example, it may work in some architectures ('components', library elements), while the assignment of copyright is not usually a practical solution.

Mr Dalmas shared some lessons about contractual agreements. It is important to give all the necessary rights, although always within limits, since changes in the spin-off management can occur very rapidly (people, ownership); in particular, exclusivity should always be limited (domain, time or revenue based).

Another important suggestion is to fix the return, as a percent, on the total turnover. Software grows old and expectations should be realistic: the counterpart will most likely demand a decrease in the royalties due over time (this might even stop at some point). As a result, the option to 'sell' the spin-off should be considered as part of the exit strategy as it is in fact a desirable outcome for the spin-off. In this respect, stocks and stock options can provide success fees.

Last but not least, like the other speakers before him, Mr Dalmas warned about always being careful with 'bundles' – several pieces of software licensed as a whole – to avoid any breach of copyright.

He concluded by stating that the best attitude to have might be to never expect anything from the spin-off other than focusing on its own business. It is impossible to anticipate everything, so the focus should be on building trust and making extensive use of amendments to reshape plans.

Conclusions

Software commercialisation was of particular interest from the business model point of view. In fact, at a very high level, companies can opt to create a spin-off, or to follow the licensing route, either as an open-source or a proprietary licence.

The various contributors, with their diverse backgrounds in the legal, technical and commercial professions, offered multiple perspectives on the factors to be considered when commercialising software. Different best practices were shared and important lessons learned emerged, which will certainly serve the community of technology transfer practitioners well.

There were a number of discussions on open source vs. proprietary software, on reward mechanisms for researchers, on how to encourage entrepreneurship, as well as good management practices. Staff retention was another important point raised, as was how to ensure that a research institute stimulates entrepreneurship without losing human capital.

One of the most recurring messages concerned the importance of auditing the software to avoid any breaching of an existing licence agreement due to copyleft mechanisms. This is a simple measure that can preserve a product's commercial value.

Throughout the workshop, participants explored the possibility for direct collaborations between academic researchers and policy-makers, in particular in relation to two matters: patenting vs. copyrights, and commercialisation routes. There were many discussions on whether software products with no connected hardware applications should only be copyrighted or should also be patented, as they are in the USA. Opinions differed, and the pros and cons where evaluated, but in general it was agreed that this would be an important point to discuss from a policy point of view.

Also discussed was how a more general policy approach could be considered to formalise the path to software commercialisation.

Various participants engaged in discussions on possible collaborations, and several new projects may result from this.

List of abbreviations and definitions

CEO – Chief Executive Officer

CTO – Chief Technical Officer

EU – European Union

EUR – Euro

GNU – A free software operating system

GPL – General public license

IP – Intellectual property

IPR – Intellectual property rights

JRC – Joint Research Centre

OS – Open source

OSS – open source software

PRO – Public research organisation

SaaS – Software as a service

SLA – Service level agreement

TTO – Technology transfer office

USD – US Dollar

Workshop Agenda

2nd TTO Circle Workshop on Best Practices

"Software as an asset for technology transfer"

29th September 2015

Brussels CDMA

Chairing: Wouter VERBURG, *Deputy Head of Unit, IP and TT, European Commission, DG JRC*

11:30 – 13:30	**Registration & Welcome Lunch**
13:30 – 13:45	**Welcome Address**
	Introduction to the workshop, *Wouter VERBURG, Deputy Head of Unit, IP and TT, European Commission, DG JRC*
13:45 – 14:45	**General Presentation - Setting the Scene**
13:45 – 14:05	*Jean-Pierre LORRE, Linagora* **Public research software dissemination – a view from the open source industry**
14:05 – 14:25	*Folkert TEERNSTRA, TNO* **Software start-up company – The experience from a public research spin-off**
14:25 – 14:45	*Asier RUFINO, Tecnalia* **Software start-up company – The experience from a public research spin-off**
	Moderator: Jean-Paul TRIAILLE (JRC); Rapporteur: Bernard DENIS (CERN)

14:45 – 15:15	**Coffee break**
15:15 – 16:30	**Session 1 : Forms of Software Protection**
15:15 – 16:35	*Jean Paul TRIAILLE, IP and TT, European Commission, DG JRC* **Copyright as a form of protection for Software**
15:35 – 15:55	*Frank VAN COPPENOLLE, Gevers* **Patents as a form of protection for Software (incl. comparison EU vs. US)**
15:55 – 16:15	*David MAZUR, CERN* **Copyright or patent: what implications for licence agreements?**
	Moderator: Stefano GENTILE (JRC); Rapporteur: Bernard DENIS (CERN)
19:00	**Dinner – Stanhope Hotel**

30th September 2015

Brussels CDMA

Chairing: Wouter VERBURG, *Deputy Head of Unit, IP and TT, European Commission, DG JRC*

08.30 – 09:00	**Welcome Coffee**
09:00 – 10:10	**Session 2 : Open Source**
09:00 – 09.25	*Jean-Paul TRIAILLE, IP and TT, European Commission, DG JRC* **Open source mechanisms and concepts**
09:25 – 09:45	*Benjamin JEAN, Inno3* **Open source strategies and practices**
09:45 – 10:05	*Stephane DALMAS (INRIA)* **Auditing issues and tools**
10:05 – 10:25	*Stefano GENTILE, IP and TT, European Commission, DG JRC* **Risks and compatibility between licenses**
	Moderator: Bernard DENIS (CERN); Rapporteur: Michel NEU (CEA)

10:30 – 11:00	**Coffee Break**

11:00 – 10:45	**Session 3: Business Models and Spin-offs**
11:00 – 11:25	*Jérémie FAYS, TTO, University of Liège* **Dissemination strategy**
11:25 – 10:45	*Arnaud LAPREVOTE, INRIA* **Business models**
11:45 – 12:05	*Stéphane DALMAS, INRIA* **Support to spin-off**
	Moderator: Annarita FERRERI (JRC); Rapporteur: Jean-Paul TRIAILLE (JRC)

12:15 – 13:15	**Lunch**
13:15 – 13:30	**Conclusions** **AOB**

Participant List

Bernard DENIS	CERN
Nick ZIOGAS	CERN
David MAZUR	CERN
Michel NEU	CEA
Alberto SILVANI	CNR
Erwan HUON DE KERMADEC	EARTO
Jean Michel CHAIZE	ESRF
Emanuel WEBER	ETH
Frank VAN COPPOLE	GEVERS
Nikolaos MINAS	GEVERS
Frederic DE CONICK	GEVERS
Kathleen DE BELDER	IMEC
Ilse HEYVAERT	IMEC
Amruta RAMSING TIRMALE	IMEC
Benjamin JANE	INNO3
Stephane DALMAS	INRIA
Arnaud LEPREVOTE	INRIA
Mattias DINNETZ	JRC
Alessandro FAZIO	JRC
Annarita FERRERI	JRC
Stefano GENTILE	JRC
Sergio GRANDE	JRC
Karolina GUTT-MOSTOWY	JRC
Jean-Paul TRIALLIE	JRC
Wouter VERBURG	JRC
Alea LOPEZ de SAN ROMAN	LERU
Jean-Pierre LORRE	LINAGORA
Bernd Ctortecka	MAX PLANCK
Asier RUFINO	TECNALIA
Asier SESMA	TECNALIA
Folkert TEERNSTRA	TNO
Jeremie FAYS	UNIVERSITE' DE LIEGE

KJ-NA-28094-EN-N

JRC Mission

As the science and knowledge service of the European Commission, the Joint Research Centre's mission is to support EU policies with independent evidence throughout the whole policy cycle.

EU Science Hub
ec.europa.eu/jrc

@EU_ScienceHub

EU Science Hub - Joint Research Centre

Joint Research Centre

EU Science Hub

Publications Office

doi:10.2760/716557

ISBN 978-92-79-61701-0